4.3 Requirements Management

4.3.1 Introduction to Requirements Management

The Requirements Management process, an element of System Engineering (SE), is an activity that spans the program's entire lifecycle. It is associated with iterative identification and refinement, to successively lower levels, of the top-level requirements, functional baselines and architectures, and synthesis of solutions established for the preferred system concept. For the purposes of Requirements Management, a system or a product shall mean any physical product being designed, developed, and/or produced, or any intangible product such as the development of a process or service-based product.

The Requirements Management process defines, collects, documents, and manages all requirements, including the complete requirements set consisting of the Mission Need Statement (MNS), the initial Requirements Document (iRD) and final Requirements Document (fRD), and the system and procurement specifications. A requirement is defined as a condition or capability that shall be met or exceeded by a system or a component to satisfy a contract, standard, specification, or other formally imposed document. Executing this process results in the authorized, organized, and baselined set of requirements for the product. These requirements are presented as requirements sets, usually in the form of requirements documents, to all other applicable SE and Federal Aviation Administration (FAA) processes. To effectively develop and manage system requirements, all requirements shall be developed through this process.

4.3.1.1 Process Description

4.3.1.1.1 Purpose

Requirements Management's purpose is to establish a layered approach that defines the necessary and sufficient attributes of the lower-level system components required for the product's successful development, production, deployment, operation, and disposal. Successful completion of this process is measured by the acceptable transformation of stakeholder needs into discrete, verifiable, low-level requirements. The process identifies, clarifies, balances, and manages the entire requirements set through interactive dialogue with all stakeholders. The top-level process appears in Figure 4.3-1.

4.3.1.1.2 Requirements Management Objectives

Requirements Management is an iterative process that:

- Identifies and captures the requirements applicable to the system
- Analyzes and decomposes the requirements into clear, unambiguous, traceable, and verifiable requirements
- Allocates the requirements to the appropriate component within the system hierarchy and/or to the appropriate organizational entities
- Derives lower-level requirements from higher-level requirements in the system hierarchy
- Establishes the method of verification for each requirement
- Ensures that the product complies with the requirements

- Manages, documents, and controls the requirements and changes to them in a traceable manner

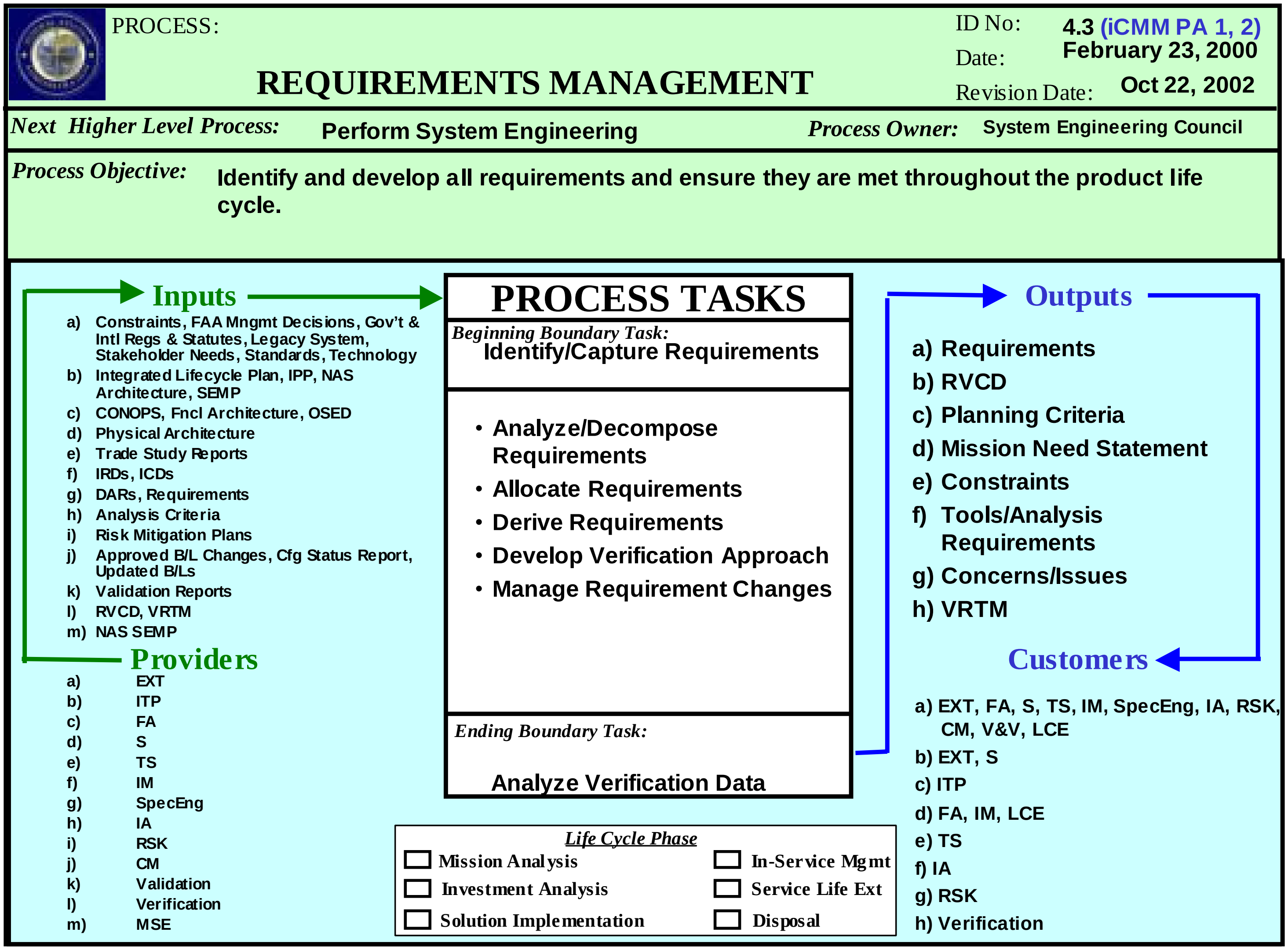


Figure 4.3-1. Requirements Management Process-Based Management Chart

4.3.1.2 Management

The Requirements Management process bridges integrated product development system stages. The products of this process are baselined in accordance with the milestones established in the Integrated Program Plan (IPP) for the applicable project. Prerequisites for successful performance of the process are:

- Empowering a requirements analysis team with the authority and mission to execute the process
- Assigning an experienced team leader knowledgeable in SE principles and committed to the standard SE methods documented herein

- Assigning team members that are experienced and knowledgeable in relevant engineering, manufacturing, operational, specialty engineering, and support disciplines
- Establishing the criteria for decisionmaking and any supporting tools
- Completing the relevant training of team members in using this process and relevant tools
- Defining the formats of the output deliverables from this activity

4.3.2 Inputs to Requirements Management

An input to the Requirements Management process is defined as information received during the process. Inputs are classified according to their source (i.e., external or internal). External inputs come from sources outside SE. Internal inputs come from other SE processes as described in this manual. Typical inputs include Stakeholder Needs and objectives, missions, measures of effectiveness (MOE) and measures of suitability, environments, key performance parameters, technology base, output requirements from prior application of SE, and program decision requirements. Input requirements shall be comprehensive and defined for both system products and system processes, including the eight lifecycle functions of development, manufacturing, verification, deployment, operations, support, training, and disposal.

Requirements Management is an iterative process that flows from a high level to a low level of requirements. Therefore, some of the inputs described in the following paragraphs may be inputs to one stage of the requirements development process and outputs of other stages. All requirements sources described were, at one point in the process, inputs and shall be captured. The inputs to the Requirements Management process are as follows.

4.3.2.1 External Inputs

External inputs come from outside SE's boundaries.

4.3.2.1.1 Constraints

A Constraint is a boundary condition within which the system remains while satisfying the aggregate system requirements.

4.3.2.1.1.1 External Constraints

External constraints, including guidelines and assumptions, shall be identified. External constraints are imposed from outside the project or system boundaries. External conditions under which the mission is to be performed and systems developed are described. The conditions may include cost, schedule, performance, technology, use of infrastructure, labor/management agreements, and programmatic constraints. Additional assumptions concerning programmatics, technology, and environments that may be required are captured.

4.3.2.1.1.2 Internal Constraints

Internal constraints, including assumptions, guidelines, and program-specific constraints, shall be identified. Internal constraints are imposed from within the project or system boundaries but outside of the SE process boundary. Program-specific conditions under which the mission is to be performed and systems developed are described. The conditions may include cost, schedule, performance, technology, use of infrastructure, and programmatic constraints.

Additional assumptions concerning programmatics, technology, and environments that may be required are captured.

4.3.2.1.1.3 Program-Specific Constraints

Program-specific organizational constraints and assumptions are captured, as well as program-specific needs, schedule constraints, and events.

4.3.2.1.1.4 Technology Constraints

Technology availability or technology constraints are captured. Technology necessary to satisfy requirements and the resulting derived requirements are described. Constraints identify the envelope of the technology operation. These inputs may include identifying key technologies, performance, maturity, cost, and risks; they may also include technology breakthroughs and forecasts.

4.3.2.1.2 Standards, Specifications, and Handbooks

Specified government standards, external standards, and general specifications or handbooks to be employed on the program are identified. The most common standards, specifications, and handbooks used in FAA requirements documents appear in Appendix A.

4.3.2.1.2.1 Standards

A standard is a document that establishes engineering and technical requirements for processes, procedures, practices, and methods that have been adopted as standard. Standards may also establish requirements for selection, application, and design criteria for material. The FAA, Department of Defense (DoD), other U.S. Government agencies, the RTCA, international organizations, and commercial standards organizations publish standards.

4.3.2.1.2.1.1 RTCA Standards

The RTCA publishes standards as Minimum Operational Performance Standards (MOPS) and Minimum Aviation System Performance Standards (MASPS).

4.3.2.1.2.1.1.1 Minimum Operational Performance Standards

The MOPS contain performance requirements for avionics. The standards describe typical equipment applications and operational goals and establish the basis for required performance and test procedures for verification under a common set of standards. Definitions and assumptions essential to proper understanding are provided, as well as installed equipment tests and operational performance characteristics for equipment installations. The MOPS also provide information that explains the rationale for equipment characteristics and stated requirements.

4.3.2.1.2.1.1.2 Minimum Aviation System Performance Standards

The MASPS address the user-level service requirements used to qualify the system for operational acceptance and to allocate requirements for the subsystems (including avionics). The standards provide information that explains the rationale for system characteristics, operational goals, requirements, and typical applications.

4.3.2.1.2.2 Specifications

A specification is a document prepared specifically to support an acquisition that clearly and accurately describes the essential technical requirements for purchased material or products and the criteria for determining whether the requirements are satisfied. The FAA, DoD, other U.S. Government agencies, international organizations, and commercial standards organizations publish specifications.

4.3.2.1.2.3 Handbooks

A handbook is a guidance document that contains information or guidelines for use in design, engineering, production, acquisition, and/or supply management operations. These documents present information, procedural and technical use data, or design information related to processes, practices, services, or commodities. Handbooks provide industry with reference materials that help to standardize FAA assets. Use of handbooks is optional unless required by a specification or contract document. The FAA, DoD, other U.S. Government agencies, international organizations, and commercial standards organizations publish handbooks.

4.3.2.1.2.4 Federal Aviation Administration Orders

An FAA order is a permanent directive on individual subjects or programs that apply to the FAA. It directs action or conduct using action verbs. Orders also prescribe policy, delegate authority, and empower and/or assign responsibility for compliance with stated requirements or direction. Orders empower or direct only FAA personnel and carry no weight with contractors. Thus, orders shall not be used in contract documents. They are not referenced in requirements documents but are used as inputs with the potential to generate requirements.

4.3.2.1.2.5 National Airspace System Management Directives

NAS-MD-001, "National Airspace System Master Configuration Index," lists all baselined equipment and software currently operational or under procurement for the National Airspace System (NAS) with current approved baseline documentation. FAA and contractor personnel use NAS-MD-001 to identify configuration items and documentation requiring NAS Change Proposals (NCP).

4.3.2.1.3 Federal Aviation Administration Management Decisions

Management decisions that are imposed on the system from the national, department, or agency level are captured.

4.3.2.1.4 Government Policy

4.3.2.1.4.1 Government Regulations and Statues

Government statutes and military and civilian regulations impacting the system are identified, including requirements incorporated into legislation (e.g., safety or security requirements). These requirements also include government standards that have been mandated as part of a contract.

4.3.2.1.4.2 International Policy

The International Civil Aviation Organization (ICAO) develops and publishes international Standards and Recommended Practices (SARP). A standard is any specification for physical characteristics, configuration, material performance, personnel, or procedure that is applied uniformly for the safety or regularity of international air navigation and to which the international aviation community conforms. A recommended practice is identical to a standard except that it is not considered necessary—only desirable.

4.3.2.1.4.3 Federal Aviation Administration Policy

This category covers all FAA agency-wide management decisions and policy requirements imposed by FAA agency-wide mandate. The category may include technical, operational, acquisition, financial, and other requirements. FAA policy is invoked using the FAA Directives System, as described in FAA Order 1320.1, “FAA Directives System.”

4.3.2.1.4.4 Acquisition Management System Limitations

New or revised directions and limitations established by the Acquisition Management System (AMS) are identified.

4.3.2.1.5 Legacy Systems

Requirements from past and current systems are captured and analyzed for applicability.

4.3.2.1.6 Stakeholder Needs

4.3.2.1.6.1 National Airspace System Concepts of Operations Document

The NAS Concepts of Operations (CONOPS) document provides a CONOPS from the perspectives of NAS users and service providers. It is the basis for an incremental benefits-driven approach toward NAS evolution. The document is arranged in a phases-of-flight approach, including Flight Planning, Surface, Arrival/Departure, En Route, and NAS Management. It is the source document for all NAS operational requirements.

4.3.2.1.6.2 Mission Need Statement

The MNS is the first document to translate the NAS CONOPS into the needs and requirements of the users and service providers. It identifies the decision factors relevant to a capability shortfall or a technological opportunity to satisfy a mission more efficiently or effectively. The MNS justifies, in rigorous analytical terms, the need to resolve a shortfall in services required by its users and service providers or to explore a technological opportunity for more efficient and effective mission performance. The MNS identifies the mission area, needed capability, current capability, capability shortfall, impact to users and service providers if the shortfall is not resolved, benefits, timeframe for resolving the shortfall, criticality of the mission, and resource estimate.

4.3.2.1.6.3 Operational Scenarios

Operational scenarios provided by the user describe how the CONOPS is implemented. They may be incorporated into the MNS or provided as a separate document.

4.3.2.1.6.4 Requirements Document

The requirements document establishes the operational framework and performance baseline, traces Functional Analysis to the NAS CONOPS and the MNS, and is the primary source document for the system requirements. This document is the principal force driving the search for a realistic and affordable solution to the mission need. The iRD is developed early in the process by the sponsoring organization. It translates the need in the MNS into initial top-level requirements that address concerns such as performance, supportability, physical and functional integration, human integration, security, test and evaluation, implementation and transition, quality assurance, configuration management, and in-service management. The iRD does not describe a specific solution to a mission need. It is recommended that the iRD not preclude leasing, commercial, or non-development solutions. The fRD defines exactly the operational concept and requirements that are to be achieved and is the basis for evaluating the readiness of resultant products and services to become operational.

4.3.2.1.7 External Interface Studies

System external interface studies and analyses that characterize and define the interfaces between the system and external environment are reviewed or conducted. These studies identify functional and physical characteristics between two or more elements that are provided by different agencies, as well as resolve problems. Topics include issues, option assessments, impact assessments, interfaces and connections, interferences, and configuration options.

4.3.2.1.8 National Airspace System Architecture

The NAS Architecture is a strategic and evolutionary plan for modernizing the NAS that supports investment analysis tradeoffs. It focuses on defining and delivering the services that meet aviation industry and public needs, which it accomplishes by decomposing the services into capabilities that are the functions and activities necessary to deliver a service. Each capability is defined by the implementations steps required to deliver the capabilities. Each implementation step is defined in terms of the mechanisms required to provide each step. Finally, each mechanism is defined in terms of the people, systems, and support activities provided by the procuring office. The NAS Architecture presents a comprehensive design that shows each major mechanism within the NAS, including interfaces and data flows. Use of a documented design, complete with traceable requirements, as the foundation for the architecture not only provides a complete picture of the NAS but also provides a roadmap for implementing future enhancements.

4.3.2.1.9 National Airspace System System Engineering Management Plan

The NAS System Engineering Management Plan (NAS SEMP) defines the relationship between the NAS SE levels, including requirements management, and the roles and responsibilities of each level. The SE levels are defined in the NAS SEMP and include the Enterprise, Domain, and Functional levels.

4.3.2.1.10 National Airspace System Requirements

4.3.2.1.10.1 NAS Systems Requirements Specification (NAS-SR-1000)

This FAA document defines the operational requirements and is the approved document for operational requirements for the NAS. The document serves as a basis to perform studies and analysis and to identify engineering concepts to satisfy operational requirements. It also serves as a source document for system specification preparation.

4.3.2.1.10.2 NAS Design Specification (NAS-DD-1000).

This FAA document defines the functional architecture, including basic NAS elements, sub-elements, subsystems, and their interrelationships.

4.3.2.1.10.3 NAS System Specification (NAS-SS-1000).

This FAA document defines functional, performance, design, construction, logistics, personnel and training, documentation, verification, and interface requirements for the NAS.

4.3.2.2 Internal Inputs

Internal inputs come from inside SE's boundaries.

4.3.2.2.1 Technical Planning

4.3.2.2.1.1 Integrated Program Plan

The Requirements Management planning section of the IPP (Integrated Technical Planning (Section 4.2)) specifies the tasks, products, responsibilities, and schedules needed to manage requirements throughout product development. It details the total work effort for managing requirements. This work includes "Task 1: Identify and Capture Requirements" (Paragraph 4.3.3.1); "Task 2: Analyze and Decompose Requirements" (Paragraph 4.3.3.2); "Task 3: Allocate Requirements" (Paragraph 4.3.3.3); "Task 4: Derive Requirements" (Paragraph 4.3.3.4); and "Task 6: Manage Requirements Changes" (Paragraph 4.3.3.6).

4.3.2.2.2 Functional Analysis

4.3.2.2.2.1 Concept of Operations

A Concept of Operations (CONOPS), which is a user-oriented document that describes a proposed system's functional requirements from the user's viewpoint, is obtained from the Functional Analysis process (Section 4.4). The CONOPS document is written to communicate overall quantitative and qualitative system characteristics to the user, buyer, developer, and other organizational elements. The CONOPS aids in requirements capture and communicates the need to the developing organization. The CONOPS describes the existing system, current environment, users, interactions among users and the system, and organizational impacts. A CONOPS is essentially a top-level narrative Functional Analysis and is the basis for developing the MNS.

4.3.2.2.2.2 Functional Architecture

Every function required to satisfy a system's operational needs shall be identified and defined. Once defined, the functions are used to define system requirements, and a Functional Architecture is developed based on the identified requirements. The process is then taken to a greater level of detail, as the identified functions are further decomposed into subfunctions, and the Functional Architecture and requirements associated with those functions are each decomposed as well. This process is iterated until the system has been completely decomposed into basic subfunctions, and each subfunction at the lowest level is completely, simply, and uniquely defined by its requirements. In this process, the interfaces between each of the functions and subfunctions are fully defined, as are the interfaces within the environment and external systems. The functions and subfunctions are arrayed in a Functional Architecture to show their relationships and internal and external interfaces.

The Functional Architecture includes a definition of the functions that the system needs to perform and is developed into Primitive Requirements Statements (PRS). “Task 2: Analyze and Decompose Requirements” (Paragraph 4.3.3.2) of the Requirements Management process develops these PRSs into Mature Requirements Statements (MRS).

4.3.2.2.2.3 Operational Services and Environmental Description

The Operational Services and Environmental Description (OSED) is a complete system description that includes information on all known hardware, software, people, procedures, and ambient and operational environments in the system. It consists of everything inside and outside the system that affects system performance and that is affected by system operation or both.

The OSED is used as a source to derive lower-level requirements. It describes many system characteristics that are nonfunctional, such as environments, and that are not described in the Functional Architecture. Nonfunctional requirements are derived from the OSED in “Task 4: Derive Requirements” (Paragraph 4.3.3.4).

4.3.2.2.3 Synthesis

4.3.2.2.3.1 Physical Architecture

The Physical Architecture allocates requirements to the physical hardware and/or software during the Synthesis process (Section 4.5). If requirements conflicts are discovered during the development of the Physical Architecture, those requirements are cycled back through the Requirements Management process for evaluation, which may result in conducting a Trade Study (Section 4.6), reallocating the requirement, or deriving lower-level requirements.

4.3.2.2.4 Trade Studies

Trade Studies (Section 4.6) may be conducted within and across functions to support decisions during any stage of the system’s lifecycle. They quantify through metrics the consequences of opting for various system alternatives, traceable to stakeholder requirements that may be imposed by the requirements development process. They support allocating performance requirements and determining requirements or Design Constraints; they are also used in evaluating alternatives. Trade Studies usually result in derived requirements that are developed into MRSs in “Task 2: Analyze and Decompose Requirements” (Paragraph 4.3.3.2).

4.3.2.2.4.1 Trade Study Reports

Trade Study Reports identify requirements that are affected by the results of each Trade Study (Section 4.6). The new, changed, or derived requirements flow through the entire Requirements Management process and may result in changes to the requirements baseline.

4.3.2.2.4.2 Feasibility Assessments

The Feasibility Assessment may be conducted to assess the difficulty in achieving program goals within the Constraints. Assessment results consider various aspects, such as technical, cost, and schedule, across the lifecycle. It provides information on the expectations for success, considering identified technology development needs in view of program and mission schedule and cost constraints. It also assesses the range of costs and benefits associated with several alternatives for solving a problem.

4.3.2.2.4.3 Derived Requirements

Derived requirements ("Task 4: Derive Requirements" (Paragraph 4.3.3.4)) may be developed through Trade Studies (Section 4.6) and not provided by external sources, such as the user, service provider, or government agencies.

4.3.2.2.5 Interface Management

The inputs from Interface Management (Section 4.7) identify, describe, and define interface requirements to ensure compatibility between interrelated systems and between system elements.

4.3.2.2.5.1 Interface Requirements Document

The Interface Requirements Document (IRD) defines requirements associated with external physical and functional interfaces between the particular system and other associated system(s).

4.3.2.2.5.2 Interface Control Document

The Interface Control Document (ICD) is a design document that describes the detailed, as-built implementation of the functional requirements contained in the IRD.

4.3.2.2.6 Specialty Engineering

Specialty Engineering (Section 4.8) defines and evaluates a system's specific areas, features, or characteristics. Specialty Engineering supplements the design process by defining these characteristics and assessing their impact on the program. Specialty Engineering studies often find characteristics that create a need for new or different requirements or a conflict between two or more requirements. The Specialty Engineering process develops the new or changed requirements, which become inputs to the Requirements Management process.

4.3.2.2.6.1 Design Analysis Reports

Design Analysis Reports (DAR), which document the results of a specific Specialty Engineering analysis with rationale, are inputs to the Requirements Management process. Each DAR contains a description of the system's special characteristics, a list of existing requirements that have undergone the Validation and Verification process (Section 4.12), residual risks, and candidate requirements found as a result of the analysis.

The rationale supplementing the DARs includes the scope, ground rules, assumptions, constraints, methods, and tools applicable to the analysis.

4.3.2.2.6.2 Derived Requirements

The Specialty Engineering process (Section 4.8) provides analysis that typically defines, validates, or verifies requirements. Occasionally, the analysis discovers system characteristics that are not adequately specified in the existing specification or requirements documents. When such discoveries occur, Specialty Engineering defines the necessary requirements that are consistent with the area of Specialty Engineering and the requirements standards described in Requirements Management.

4.3.2.2.7 Integrity of Analysis

4.3.2.2.7.1 Analysis Criteria

If the Requirements Management process requires an analysis or selection of a tool, Analysis Criteria for that analysis or selection are captured. The Analysis Criteria for conducting a required analysis is contained within the Analysis Management Plan.

4.3.2.2.8 Risk Management

4.3.2.2.8.1 Risk Mitigation Plans

Concerns/Issues identified by any SE process are analyzed in the Risk Management process (Section 4.10). Risk Mitigation Plans that result from risk analysis become inputs to the Requirements Management process. Requirements that present a risk are processed through the Requirements Management process for reanalysis, reallocation, and rederivation, as needed.

4.3.2.2.9 Configuration Management

4.3.2.2.9.1 Approved Baseline Changes

Approved changes to the baselined requirements set are captured from the Configuration Management process (Section 4.11). “Step 6: Manage Requirements Changes” (Paragraph 4.3.3.6) inserts the Approved Baseline Changes into the requirements set.

4.3.2.2.9.2 Configuration Status Reports

Configuration Status Reports are captured from the Configuration Management process (Section 4.11). “Step 6: Manage Requirements Changes” (Paragraph 4.3.3.6) uses these reports to maintain a status accounting of all requirements.

4.3.2.2.9.3 Updated Baselines

Updated Baselines are captured from the Configuration Management process (Section 4.11). “Step 6: Manage Requirements Changes” (Paragraph 4.3.3.6) controls the updated baseline configuration.

4.3.2.2.10 Validation

The Validation process (Section 4.12) determines if the requirements produced by the Requirements Management process are sufficiently correct and complete. Requirements that are not validated are captured and resubmitted to the Requirements Management process.

4.3.2.2.10.1 Validation Report

The Validation Report summarizes the results of the Validation process (Section 4.12) and communicates the Validation Table to the Requirements Management process.

The Validation Report contains:

- Summary of validation results
- Description of the system and program
- Validation methodology used
- Unvalidated requirements
  - List of nonconforming requirements
  - Recommendations for correction of nonconforming requirements
- Validation Table
- Discussion of trends and patterns of failure, evidence of systemic failings, and emerging threats to system services.

4.3.2.2.10.2 Validation Table

The Validation Table is a listing of all requirements that describes if a requirement has been validated, where the requirement may be found, source of validation, corrective action to be taken if necessary, and the corrective action owner. Table 4.12-1 in Validation and Verification (Section 4.12) is an example of a Validation Table. The completed Validation Table is included in the requirements document and is the basis for the Verification process.

4.3.2.2.11 Verification

The Verification process (Section 4.12) determines that applicable requirements are satisfied by the design solution.

4.3.2.2.11.1 Verification Requirements Traceability Matrix

The Validation Table from the Validation process (Section 4.12) is further refined into a Verification Requirements Traceability Matrix (VRTM), the heart of the Verification process. The strategy or method used to verify each requirement is specified in a Verification Requirement, and the Verification Requirements are listed in the VRTM. The VRTM defines how each requirement (functional, performance, and design) is to be verified, the stage in which verification is to occur, and the applicable verification levels. The VRTM establishes the basis for the verification program. The VRTM is initiated by the Requirements Management process, which sends it to the Verification process, which returns it to Requirements Management when verification has been completed.

4.3.2.2.11.2 Requirements Verification Compliance Document

The Requirements Verification Compliance Document (RVCD) provides evidence of compliance for each requirement at all levels and to each VRTM requirement. The flowdown from the requirements documents to the VRTM completes the full requirements traceability. Compliance with all requirements ensures that the system-level requirements have been met. The RVCD defines, for each requirement, the verification methods and corresponding compliance information. The results of the Verification process (Section 4.12), including evidence of completion, are recorded and documented in the RVCD. It is recommended that the RVCD contain information regarding the results of each verification activity, as well as a description and disposition of conformance, nonconformance, conclusions, and recommendations. Compliance information provides either the actual data or a reference to the location of the actual data that shows compliance with the requirement. The document also includes a section that details any noncompliance. It is recommended that this section also specify appropriate reverification procedures. The Requirements Management process captures noncompliant requirements, leading to a decision on disposition of the noncompliant requirement.

4.3.3 Requirements Management Process Tasks

The following tasks are necessary to perform this process:

- Identify and Capture Requirements
- Analyze and Decompose Requirements
- Allocate Requirements
- Derive Requirements
- Establish Requirements Verification Methods
- Manage Requirements

4.3.3.1 Task 1: Identify and Capture Requirements

4.3.3.1.1 Description

The Identify and Capture Requirements activity identifies, prioritizes, and extracts all written directives, including documented stakeholder negotiations/discussions, and internally derived requirements that are relevant to the particular stage of the system lifecycle. This activity is performed on the entire system, including any requirements that are known at this stage about how the system shall perform during its lifecycle and any constraints imposed on the system design/production by stakeholders and internal functions (i.e., manufacturing, product support, agency-level policies, suppliers). There are many different types, or categories, of requirements, as identified and defined in Paragraph 4.3.3.2.1.4.3. Requirements are typically categorized by the stage of the system lifecycle in which the requirement is obtained and by the function/user that generates the requirement. The primary objective is to consolidate baseline or approved system requirements so that they may serve as a foundation for later refinement and/or revision by subsequent functions in SE. This consolidation also allows an unambiguous and traceable flowdown of source requirements throughout the NAS Architecture as well as the product hierarchy. It is also important to negotiate with both external and internal stakeholders to reach agreement on which documents and to what level requirements need to be traced. This activity helps to ensure that the visibility stakeholders expect to obtain from requirements traceability may be achieved. This foundation needs to be as complete and accurate as possible and shall be fully traceable to the requirements source documentation.

4.3.3.1.2 Scope

The scope of the requirements set shall include sufficient specification of all the system functions and all the external interfacing systems, including the system environment. This task may require considering a wider domain than the immediate physical boundary of the product and its components. Different boundaries may need to be defined for different states, modes, and capabilities. Refinement of these boundary definitions is an iterative process that occurs as more information is discovered about the true nature of the required system functions and performance (Interface Management (Section 4.7)). In this process, hardware, software, and system requirements are analyzed and refined to ensure that they are consistent, clear, valid, feasible, compatible, complete, and verifiable and that they do not include detail design information.

4.3.3.1.3 Result

The result of performing this activity shall be a baseline set of requirements. The requirements shall be captured in an organized fashion. It is recommended the that information be readily accessible for reference by other program personnel as needed. This activity is the basis for discovering and successively refining the requirements to be recorded and maintained over the product's lifecycle.

4.3.3.1.4 Compatibility

The selected requirements methodology shall be compatible with other methodologies applied across the FAA, and the analysis methodology supported with the necessary tools, as required by the Integrity of Analysis process (Section 4.9).

4.3.3.1.5 Detailed Task 1 Description

Figure 4.3-2 describes the flow of the Identify and Capture Requirements task.

4.3.3.1.5.1 Task 1.1: Define Stakeholder Expectations

Stakeholder expectations are defined and quantified, and stakeholder expectations in the FAA come from the operational stakeholder in the form of:

- CONOPS
- MNS
- iRD or fRD

They are transformed into baselined requirements sets at a successively lower level through iteration of the Requirements Management process. It is recommended that the definition of stakeholder expectations be balanced with an analysis of their effects on the overall system design and performance as well as on human engineering; knowledge, skills, and abilities; availability; reliability; safety; and training requirements of the humans required to support lifecycle processes. Stakeholder expectations include:

- What the system is to accomplish (functional requirements)
- How well each function is to be performed (performance requirements)
- The operational and ambient environment in which the system is to be operated
- Constraints under which the system is to be developed or operated (e.g., funding, cost or price objectives, schedule, technology, nondevelopmental and reusable items, physical characteristics, and hours of operation per day)

4.3.3.1.5.2 Task 1.2: Define Project and Corporate Constraints

Project and corporate constraints that impact design solutions shall be identified and defined. The NAS Architecture may also impose long-range planning constraints through the approved capabilities and implementation steps.

4.3.3.1.5.2.1 Project Constraints

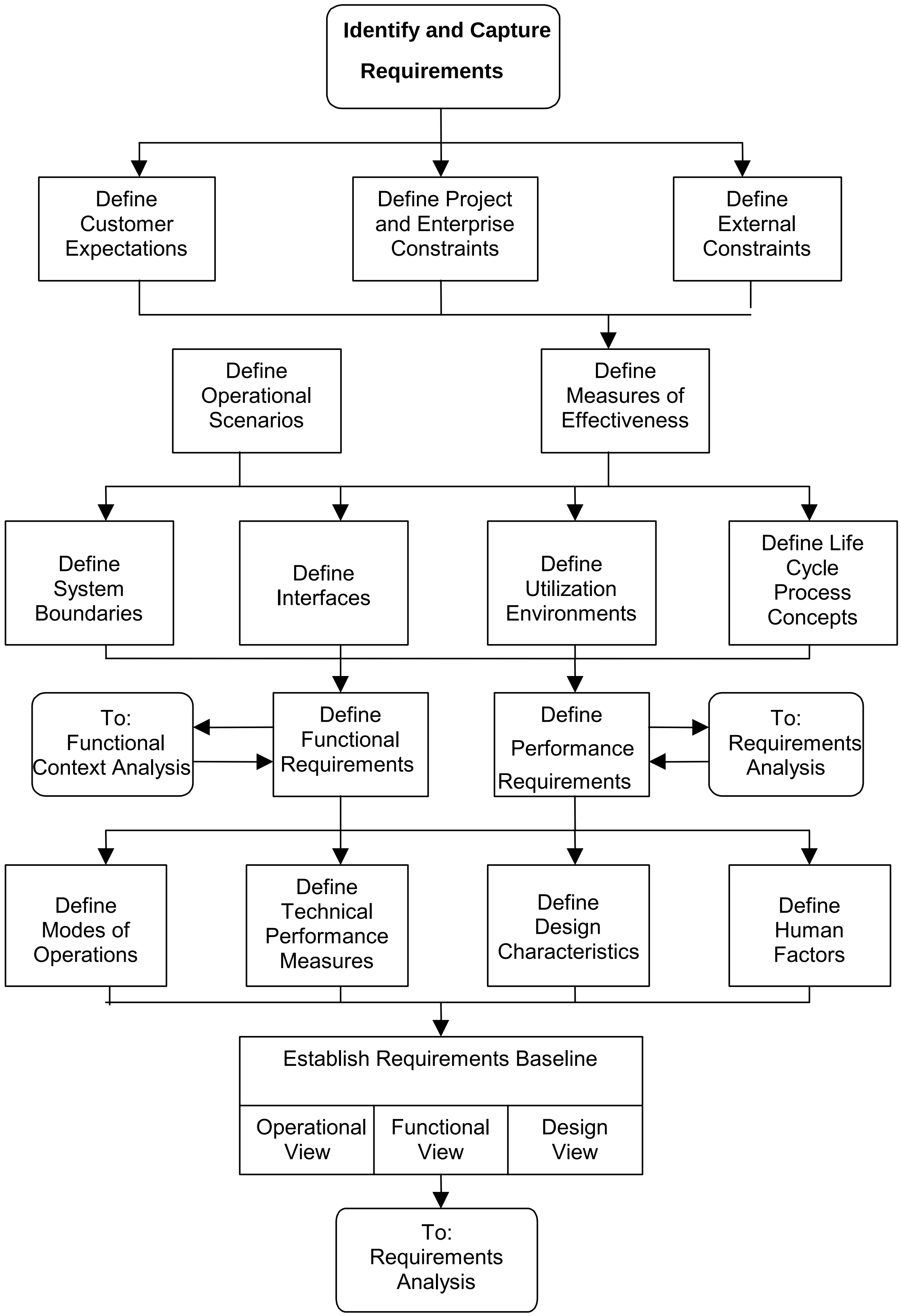
Project constraints include:

- Existing approved specifications and baselines
- Updated NAS Architecture implementation steps
- Updated NAS Architecture segments and mechanisms
- Availability of automated tools
- Required metrics for measuring technical progress

4.3.3.1.5.2.2 Corporate Constraints

Corporate constraints include:

- Management decisions from the Joint Resources Council or other management review
- FAA-wide general specifications, standards, handbooks, and guidelines
- FAA policy directives
- Established lifecycle processes
- Physical, financial, and human project resources



525 **Figure 4.3-2. Identify and Capture Requirements Flow**

4.3.3.1.5.3 Task 1.3: Define External Constraints

External constraints that impact design solutions or implementation of SE activities shall be identified and defined. These include:

- U.S. Government and international laws and regulations
- Industry, international, and other general specifications, standards, and guidelines
- ICAO SARPs
- RTCA MOPS and MASPS
- Human-related specifications, standards, and guidelines
- The technology base
- Interfacing systems

4.3.3.1.5.4 Task 1.4: Define Operational Scenarios

Operational scenarios that define the range of the anticipated system uses shall be identified and defined. For each operational scenario, expected interactions with the environment and other systems, human tasks and task sequences, and physical interconnections with interfacing systems and platforms shall be defined.

Data for this step comes from the CONOPS, iRDs and fRDs, and the NAS Architecture.

4.3.3.1.5.5 Task 1.5: Define Measures of Effectiveness

System effectiveness measures that reflect overall stakeholder expectations and satisfaction are defined. Key MOEs may include performance, safety, operability, usability, reliability, maintainability, time and cost to train, workload, human performance requirements, or other factors. Data for this step comes from the CONOPS, iRDs and fRDs, the NAS Architecture, the NAS Requirements, and operational scenarios.

4.3.3.1.5.6 Task 1.6: Define System Boundaries

System boundaries are defined as follows:

- System elements that are under design control and elements that are not
- Expected interactions among system elements under design control and external and/or higher-level and interacting systems outside the system boundary

Data for this step is obtained from any internal, external, policy, or technology constraints; CONOPS; MNS; iRDs and fRDs; and Functional Analysis.

4.3.3.1.5.7 Task 1.7: Define Interfaces

The functional and physical interfaces are defined to external or higher-level and interacting systems, platforms, and/or products in quantitative terms. Functional and physical interfaces may include mechanical, electrical, thermal, data, communication procedural, human-machine, and other interactions required. Interfaces may also be considered from an internal/external perspective. Internal interfaces address elements inside the boundaries established for the system; they are generally identified and controlled by the contractor responsible for developing the system. External interfaces involve entity relationships outside the established system boundaries.

Data for this step is in IRDs, ICDs, Functional Analysis, MNS, and iRDs and fRDs.

4.3.3.1.5.8 Task 1.8: Define Utilization Environments

Utilization environments for each of the operational scenarios shall be defined. All environmental factors, operational and ambient, that may impact system performance need to be identified and defined. Also identified are factors that ensure that the system minimizes the potential for human or machine errors or for failures that cause accidents or death and that impart minimal risk of death, injury, or acute chronic illness, disability, and/or reduced job performance of the humans who support the system lifecycle. Specifically, weather conditions (e.g., rain, snow, sun, wind, ice, dust, and fog); temperature ranges; topologies (e.g., ocean, mountains, deserts, plains, and vegetation); biological factors (e.g., animal, insects, birds, and fungi); time (e.g., day, night, and dusk); induced factors (e.g., vibration, electromagnetic acoustic, x-ray, and chemical), or other environmental factors are defined for possible locations and conditions conducive to system operation. It is recommended that effects on hardware, software, and humans be assessed for impact on system performance and lifecycle processes.

Data for this step is contained in the OSED, Trade Studies, Specialty Engineering analysis, and FAA and Military Standards, Specifications, and Handbooks.

4.3.3.1.5.9 Task 1.9: Define Lifecycle Process Concepts

The outputs of Tasks 1.1 through 1.8 are analyzed to define lifecycle process requirements necessary to develop, produce, test, distribute, operate, support, train, and dispose of system products being procured.

4.3.3.1.5.9.1 Manpower

The required job tasks and associated workload used to determine the number and mix of humans who support the system lifecycle processes shall be identified and defined.

4.3.3.1.5.9.2 Personnel

The experiences, aptitudes, knowledge, skills, and abilities required to perform the job tasks that are associated with the humans who support the system lifecycle shall be identified and defined.

4.3.3.1.5.9.3 Training

The instruction education and on-the-job or team training necessary to provide humans and teams with knowledge and job skills needed to support the system lifecycle processes at the specified levels of performance are to be identified and developed.

4.3.3.1.5.9.4 Human Engineering

Human cognitive, physical, and sensory characteristics that directly contribute to or constrain lifecycle system performance and that impact human-machine interfaces shall be identified.

4.3.3.1.5.9.5 Safety

The System Safety Engineering analysis derives and identifies requirements that are designed to control the risk of identified safety hazards.

4.3.3.1.5.10 Task 1.10: Define Functional Requirements

Functional requirements for each function of the system as determined by the Functional Analysis process (Section 4.4) shall be defined, describing what the system may be able to do. The functions identified are used in Paragraph 4.3.3.1.5.11 to define how well the functions shall be performed and to establish the performance requirements. The functions identified through Functional Analysis shall be further decomposed during functional decomposition to provide a basis for identifying and assessing design alternatives. All system requirements shall involve a functional and performance aspect, which views system requirements as having both functional and performance aspects that ensure that requirements are complete, consistent, and verifiable.

4.3.3.1.5.11 Task 1.11: Define Performance Requirements

Performance requirements for each system function shall be defined. Performance requirements describe how well functional requirements shall be performed to satisfy the MOEs. These performance requirements are the MOPS that are allocated to subfunctions during functional decomposition analysis and that are the criteria against which design solutions (derived from Synthesis (Section 4.5)) are measured. There are typically several MOPS for each MOE, which bound the acceptable performance envelope.

4.3.3.1.5.12 Task 1.12: Define Modes of Operation

The system modes of operation (e.g., full system, emergency, training, and maintenance) are defined for the system being procured. The conditions (e.g., environmental, configuration, and operation) that determine the modes of operation are defined.

Data for this step shall come from the NAS or system-level CONOPS, MNS, OSED, or Functional Analysis.

4.3.3.1.5.13 Task 1.13: Define Technical Performance Measures

Technical Performance Measures (TPM) are defined that describe the key indicators of system performance. It is recommended that selection of TPMs be limited to critical MOPs that, if not met, put the project at cost, schedule, or performance risk. Specific TPM activities are integrated into the System Engineering Master Schedule to periodically determine achievement to date and to measure progress against a planned value profile.

Data for this step comes from the CONOPS or the MNS.

4.3.3.1.5.14 Task 1.14: Define Design Characteristics

Required design characteristics (e.g., color, texture, size, anthropometrical limitations, weight, and buoyancy) are identified and defined for the system being procured. Design characteristics that are constraints and which may be changed based on tradeoff analysis (Synthesis (Section 4.5)) are identified.

Data for this step comes from the CONOPS, MNS, OSED, Functional Analysis, Tradeoff Studies, and FAA and Military Standards, Specifications, and Handbooks.

4.3.3.1.5.15 Task 1.15: Define Human Factors

Human factor considerations (e.g., design space limits, climatic limits, eye movement, reach ergonomics, cognitive limits, and usability) are identified and defined that affect operation of the system being procured. Human factors that are constraints and may be changed based on tradeoff analysis are identified.

Data for this step comes from the CONOPS, MNS, OSED, Functional Analysis, Tradeoff Studies, Specialty Engineering analysis, and FAA and Military Standards, Specifications, and Handbooks.

4.3.3.1.5.16 Task 1.16: Establish Requirements Baseline

The output of Tasks 1.1 through 1.15 forms a requirements baseline that establishes the characteristics of the system problem to be solved. Three views—operational, functional, and design—are used to define the baseline. The Operational View describes how the system products serve users. It establishes who operates and supports the system and its lifecycle processes and how well and under what conditions the system is to be used. The Functional View describes what the system does to produce the desired behavior described in the Operational View and provides a description of the methodology used to develop the view and decision rationale. The Design View describes the design consideration of the system development and established requirements for technologies and for design interfaces among equipment and among humans and equipment. The content of these views may include the information discussed in the following paragraphs.

4.3.3.1.5.16.1 Operational View

The Operational View addresses how the system serves its users. It is useful when requirements are being established that describe how well and under what condition the system is to be used. It is recommended that Operational View information be documented in an operational concept document that identifies:

- Operational need description
- Results of system operational analyses
- Operational sequences/scenarios, including utilization environments and MOEs and how the system may be used
- Conditions/events to which system products need to respond
- Operational constraints, including MOEs
- Human roles, including job tasks and skill requirements
- Training requirements, including how humans are trained to be a part of the system and support system lifecycle processes through formal, informal, embedded, on-the-job, or other forms of training
- What operations are required to ensure safety
- The security threats that the system shall be protected against

- Lifecycle process concepts, including MOEs, critical MOPS, and already existing products and services
- Operational interfaces with other systems, platforms, humans, and/or products
- System boundaries

4.3.3.1.5.16.2 Functional View

The Functional View focuses on what the system shall do to produce the required operational behavior. It includes required inputs, outputs, states, and transformation rules. The Functional View and the Operational View are the primary sources for the MNS and the requirements documents. The functional requirements, coupled with the design requirements, described in Design View below, are the primary sources of the requirements that may eventually be reflected in the system specification. Functional View information includes:

- Functional requirements that describe what system products and lifecycle processes shall do or accomplish
- Performance requirements, including qualitative (how well), quantitative (how much, capacity), and timeliness or periodicity (how long, how often) requirements
- Functional sequences for accomplishing system objectives
- TPM criteria
- Functional interface requirements with external, higher-level, or interacting systems, platforms, humans, and/or products
- Modes of operations
- Functional capabilities for planned evolutionary growth
- Verification requirements, including inspection, analysis/simulation, demonstration, and test

4.3.3.1.5.16.3 Design View

The Design View focuses on how the system is constructed. It is key to establishing the physical interfaces among operators and equipment and technology requirements. Design View information includes:

- Previously approved specifications and baselines
- Design interfaces with other systems, platforms, humans, and/or products
- Human SE elements, including safety, training, knowledge, skills, and abilities required to accomplish system functions, and characteristics of information displays and operator controls
- Characterization of operator(s) and support personnel, including special design requirements and applicable movement or visual or workload limitations
- Characterization of information displays and operator controls
- System characteristics, including design limitation (e.g., capacity, power, size, weight); technology limitations (e.g., precision, data rates, frequency, language); inherent human limitations (e.g., physical and cognitive workload, perceptual abilities, and reach and

anthropometric limitations); and standardized end items, nondevelopmental items (NDI), and reusability requirements

- Design constraints, including project, corporate, and external constraints, that limit design solutions and/or developmental procedures
- Design capabilities and capacities for planned evolutionary growth

4.3.3.2 Task 2: Analyze and Decompose Requirements

The Functional Architecture developed in Functional Analysis (Section 4.4) is translated into Primitive Requirements Statements (PRS) that, in turn, are translated into Mature Requirements Statements (MRS) in this task.

4.3.3.2.1 Analyze Requirements

The Functional Architecture is the primary input to the Requirements Management process. A Functional Architecture describes “what” a system shall accomplish. The Functional Architecture is composed of functional flow diagrams (FFD), timeline sequence diagrams, and functional N^2 charts described in Functional Analysis (Section 4.4). The Functional Architecture is a living document that increases in level of detail along with the decomposition of requirements. It is recommended that there be a level of Functional Analysis and corresponding Functional Architecture for every level of requirements (Table 4.3-1). The Requirements Management process uses the Functional Architecture to derive PRSs.

The Requirements Management process starts with recognition of a need or shortfall in system capability and progresses in increasing detail, as shown in Table 4.3-1.

Table 4.3-1. Functional Architecture to Requirements Traceability Hierarchy

Functional Architecture	Requirements
CONOPS →	Mission Need Statement
Functional Analysis 1 →	Initial Requirements Document
Functional Analysis 2 →	Final Requirements Document
Functional Analysis 3 →	System Requirements
Functional Analysis N →	System Specification to N level

4.3.3.2.1.1 Function to Requirements Transformation

The objective of function transformation is to transform functions into the functional and performance PRSs that describe the system attributes that achieve customers’ needs.

A Functional Architecture (from Functional Analysis (Section 4.4)) is transformed into PRSs through two fundamental methods: (1) a structured analysis methodology called System Functional Requirements Analysis (SFRA) and (2) Functional Architecture Referencing (FAR).

Regardless of the method used, the result is a set of PRSs associated with the system functions.

4.3.3.2.1.1.1 System Functional Requirements Analysis

SFRA is a structured methodology for developing requirements from a Functional Architecture. It requires building a matrix of functions and system characteristics then assigning a PRS to each function/characteristic pair if one is needed. The following steps produce a list of functions for which PRSs shall be developed.

4.3.3.2.1.1.1.1 List Functions

From the Functional Architecture, the functions are listed on the vertical axis of a table, such as the example included in Table 4.3-2. A tree diagram may be used to assist creation of the function list.

4.3.3.2.1.1.1.1.1 Tree Diagrams

A tree diagram is constructed from the top down. Each subfunction is shown as a branch of the tree. Using the FFD in Figure 4.4-23 as an example, the tree diagram in Figure 4.3-3 was developed as an incomplete example of what the tree diagram might look like. A completed diagram might result in a family tree hierarchy of functions.

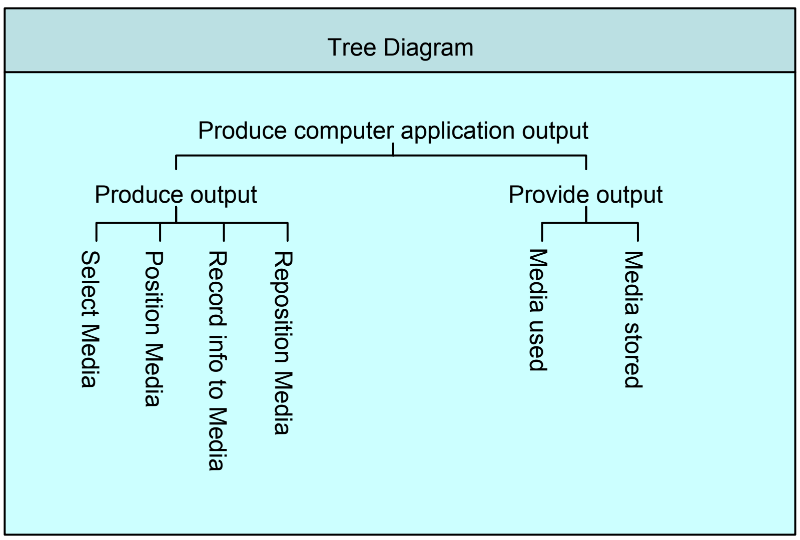


Figure 4.3-3. Tree Diagram Example

4.3.3.2.1.1.1.2 List System Characteristics

System characteristics are developed by identifying all measurable product characteristics perceived as related to meeting customer requirements. These characteristics come from (1) the external inputs described in Paragraph 4.3.2.1 and (2) analyses conducted in Specialty Engineering (Section 4.8). The characteristics include specialty requirements, constraints, standards, handbooks, management decisions, policies, and legacy requirements. The system characteristics are listed on the horizontal axis of Table 4.3-2. The specific categories and characteristics are unique to and change with each system. The material shown is for illustration only.

4.3.3.2.1.1.1.3 Determine Intersections

The purpose of this step is to determine if a need exists to translate a particular function into a PRS. If there is a significant relationship between the function and the characteristic, a PRS number is placed in that cell. “Significant” means that it was determined, using engineering judgment, that the function shall have one or more of the related characteristics in order to meet the customer’s need. Wherever there is a number, a unique PRS is required to describe that relationship. The number is associated with the unique PRS that describes the function-characteristic combination.

If it is determined that a function-characteristic combination is not significant or nonexistent, then a PRS is **not** written for that intersection.

4.3.3.2.1.1.1.4 Develop Primitive Requirements Statements

A PRS for each intersection in the table is developed in accordance with the procedure in Paragraph 4.3.3.2.1.1.3.

4.3.3.2.1.1.2 Functional Architecture Reference

This method generates PRSs from the standards, handbooks, and Specialty Engineering analyses. The functional PRSs are developed by referencing the Functional Architecture. Because of the risk of missing critical requirements, it is recommended that this method be used only when there is not enough time to perform SFRA.

4.3.3.2.1.1.2.1 Derive Primitive Requirements Statement From Standard Sources

A list of PRSs is developed. The PRSs are derived by using the sources described in Specialty Engineering (Section 4.8) and the inputs listed in Paragraph 4.3.3. The PRSs shall be developed in accordance with 4.3.3.2.1.2.

For example, assume that a reliability analysis derived a requirement that states: “Transmitter MTBF greater than 5000 op hours.” The PRS is listed as a requirement in this list. Table 4.3-3 provides an example.

Table 4.3-2. System Characteristic Matrix

Functions	Characteristics	Performance		Specialty Engineering				Environment		
		Accuracy	Thermal	Reliability	Safety	Spectrum	Oprtr workload	Radiation	Lightning	Precipitation
Detect ac state vector	Determine aircraft horizontal location	2	1		3	N	N	N	N	N
	Determine aircraft vertical location	N	N		N	N	N	N	N	N
	Determine aircraft velocity vector	N	N		N		N			
Transmit voice RF	Convert sound to high frequency (RF) signal	N	N	N		N	N	N	N	N
	Convert signal to EM wave	N	N	N	N	N		N		N
	Propagate wave through space-time					N		N	N	N
Distribute NOTAM	Encode NOTAM	N	N		N		N			
	Determine scope	N	N		N		N			
	Transmit NOTAM	N	N		N	N	N	N	N	N

Note: N = a PRS number for the specific intersection.

Table 4.3-3. Primitive Requirement Statements List

PRS Number	Primitive Requirement Statement	Functional Reference
Assign a unique number to the PRS	This is the derived PRS	Assign the PRS to a function in the Functional Architecture
126	Transmitter MTBF greater than 5000 op hours	F.3.2.1.1

4.3.3.2.1.1.2.2 Relate Primitive Requirements Statement to Functional Architecture

The Functional Architecture and existing PRSs are reviewed, and each PRS is assigned to a function in the Functional Architecture. Each requirement shall be assigned to a function, and it is recommended that each function have one or more requirements assigned to it.

4.3.3.2.1.1.2.3 Sort the Primitive Requirements Statements by Functional Reference

The list of PRSs developed in 4.3.3.2.1.1.2.2 shall be sorted or grouped so that grouped and sorted requirements allocated to an individual function are together. Table 4.3-4 is an example.

Table 4.3-4. Primitive Requirement Statements List

PRS Number	Primitive Requirement Statement	Functional Reference
126	Transmitter MTBF greater than 5000 op hours	F.3.2.1.1
34	Transmitter EMI hardened greater than 50000 volt-meters	F.3.2.1.1
212	Transmitter power less than 10 watts	F.3.2.1.2
6	Transmitted power less than or equal to table 4.3 in HERP standard 6.	F.3.2.1.2
57	Transmitted power less than or equal to table 2.1 in HERF standard 4.4.	F.3.2.1.2

4.3.3.2.1.1.2.4 Write the Functional Primitive Requirements Statement

Once requirements are sorted to functions, the functional PRSs are derived. First, the Functional Architecture used shall be appended to the requirements document. Then, for each group of PRSs, a functional PRS shall be defined in the following manner:

> **[Element] functions + as defined in + [Functional Reference (include page and figure number)]**

For the above example table, two functional PRSs are added as shown in Table 4.3-5.

Table 4.3-5. Grouped and Sorted Primitive Requirement Statements List

PRS Number	Primitive Requirement Statement	Functional Reference
126	Transmitter MTBF greater than 5000 op hours	F.3.2.1.1
34	Transmitter EMI hardened greater than 50000 volt-meters	F.3.2.1.1
220	Transmitter functions as defined in F.3.2.1.1, page A-26, figure A.2.2.	F.3.2.1.1
212	Transmitter power less than 10 watts	F.3.2.1.2
6	Transmitted power less than or equal to table 4.3 in HERP standard 6.	F.3.2.1.2
57	Transmitted power less than or equal to table 2.1 in HERF standard 4.4.	F.3.2.1.2
221	Transmitter functions as defined in F.3.2.1.2, page A-28, figure A.2.4.	F.3.2.1.2

4.3.3.2.1.1.3 Develop Mature Requirements Statements

Once the list of PRSs is developed using either SFRA or FAR, they are transformed to MRSs in accordance with Paragraph 4.3.3.2.1.3.

4.3.3.2.1.2 Primitive Requirements Statements

Requirements are first captured as a list of PRSs. A PRS is a primitive form of a requirement statement that has no punctuation or formal sentence structure and is not written in a formal specification style. The PRS form is used at this stage to improve the early requirements identification capability by removing the rigor of writing MRSs from the early concept development and to remove the considerable cost of forming mature requirements. Each PRS is uniquely numbered and follows a simple three-part format:

Name + Relation + Value

The name describes the characteristic or attribute to control; the relation details the connection between the attribute and its control value; and the value sets a quantifiable number with units or defines a standard. Numerical requirements use one of six possible relations: less than, greater than, equal to, less than or equal to, greater than or equal to, or between a range of values. For nonnumerical requirements, words such as “is,” “be,” and “conforms to” are used as the relation.

4.3.3.2.1.3 Mature Requirements Statement

Once the PRSs at any level are identified, they shall be synthesized into MRSs that satisfy the characteristics and attributes of good requirements. Requirements characteristics are the principal properties of the MRS. Characteristics may apply to individual requirements or to an aggregate of requirements. A well-defined set of MRSs needs to exhibit certain individual and aggregate characteristics. The result of performing this activity shall be a baseline set of requirements that satisfy all of the characteristics described herein and that is recorded and maintained over the lifecycle of the product, as well as accessible to all parties.

The basics of well-defined requirements are clarity, conciseness, and simplicity; elegant, entertaining prose is not needed and is undesirable. This activity describes (1) how to build requirements and (2) the essential characteristics of well-defined requirements.

An MRS is a written statement of a requirement in one or more complete sentences in a familiar language (normally English) using the idiom of a particular business sector, such as air traffic control or avionics. Normal specification standards require that the content of a specification document include complete sentences organized in a particular way. Each requirement statement shall (1) be written in proper grammar, (2) make appropriate use of standard constructs, (3) possess the characteristics and attributes of good requirements, and (4) comply to a specified standard format.

Each PRS shall be converted to specification text. A specification for a system is a published set of requirements that has been properly refined and formatted into more precise language than used for the PRSs. Usually, each PRS becomes a short paragraph when converted into specification text. A primitive requirement is connected into specification text by adding the characteristics described in the following paragraphs.

4.3.3.2.1.3.1 Paragraph Number

The type of requirements is identified, and a paragraph number is assigned according to the required format. The numbering format may be ad hoc for some requirements documents or shall adhere to a rigid format, such as a Federal Aviation Administration Acquisition System Toolset (FAST) template or FAA-STD-005 or MIL-STD-961.

4.3.3.2.1.3.2 Paragraph Title

A paragraph title is identified that is linked to the named or controlled PRS attribute.

4.3.3.2.1.3.3 Subject

The subject of the requirements is the main topic of the sentence and is linked to the named or controlled PRS attribute.

4.3.3.2.1.3.4 Directive Verb

The directive verb in the requirement sentence directs the action required and shall relate the named or controlled attribute to the value. See Paragraph 4.3.3.2.1.1.5.1.

4.3.3.2.1.3.5 Sentence Ending

The requirements sentence is ended with a period with a commonly used word or phrase that provides a reference to a standard or specification. See Paragraph 4.3.3.2.1.1.5.2.

4.3.3.2.1.3.6 Explanatory information

Explanatory, defining, or clarifying information is added after the requirements sentence if necessary to ensure understanding and avoid ambiguity.

4.3.3.2.1.3.7 Standard Constructs

Standard constructs are used to record requirements so that they possess the characteristics of good requirements.

4.3.3.2.1.3.7.1 Directive Verbs

All requirements documents shall have directive verbs that denote action, as follows:

- Use the verb "shall" to denote compulsory or mandatory action that the person being directed is obliged to take. (For example: The contractor shall furnish all facilities and equipment necessary for the tests specified herein.)
- Use the verb "may" to denote permission or an option that is not obligatory. (For example: For instruction books of 50 pages or less, multi-ring binding may be employed in lieu of saddle stitching.)
- Use the verb "will" to denote a declaration of purpose on the part of the government. (For example: The Contracting Officer will furnish shipping instructions upon request.)
- The verb "should" is not used in requirements documents. Although the word "should" is used to denote action that is recommended but not obligatory, it may imply duty or obligation in legal usage.

4.3.3.2.1.3.7.2 Commonly Used Words and Phrasings

Certain words and phrases are frequently used in requirements documents. The following rules shall apply:

- Referenced documents requirements are to be written as follows:
  - " …in accordance with Specification (or Standard)…"
  - "…shall be as specified in Specification (or Standard)…"
  - "…shall conform to…
  - "…conforming to Specification (or Standard)…"

- The phase "unless otherwise specified" shall be used to indicate an alternate course of action. The phrase shall come at the beginning of the sentence and, if possible, at the beginning of the paragraph. This phrase shall be limited in its application and used sparingly.
- The term "and/or" shall not be used in requirements documents. The following example conveys the desired meaning: "The panel shall be supported on brackets, pillars, or both."
- Do not use "minimum" and "maximum" to state limits. Use "no less than" or "no greater than." This standard construct avoids the ambiguity associated with the limiting values. This does not mean that the words "minimum" and "maximum" may not be used at all, just not to state limits.

4.3.3.2.1.3.7.3 Words and Phrases To Avoid

It is recommended that specific words and phases be avoided because they are vague, ambiguous, and general, such as “flexible,” “fault tolerant,” “high fidelity,” “adaptable,” “rapid” or “fast,” “adequate,” “user-friendly,” “support,” “maximize,” “minimize,” and “shall have the capability to.”

4.3.3.2.1.4 Characteristics of Individual Requirements

Characteristics of individual requirements may be used for requirements development as well as in requirements reviews and audits for assessing the quality of requirements. These characteristics are described below with synonyms in parenthesis.

4.3.3.2.1.4.1 Necessary

The stated requirement is an essential capability, characteristic, or quality factor of the product or process. If removed or deleted, it may cause a deficiency that is unable to be fulfilled by other capabilities of the product or process.

This is a primary characteristic, and it shall be exhibited in the requirements statement to effect a well-defined requirement. There is no room in a specification for unnecessary requirements because they add cost to the product. If a necessary requirement is deleted from the specification, a major need may not be met, even if all other requirements are satisfied.

One good test of necessity is traceability to higher-level documentation. In the case of a system specification, traceability may be verified to user documentation, such as the Operational Requirements Document. If there is no parent requirement, the requirement may not be necessary.

4.3.3.2.1.4.2 Concise (Minimal, Understandable)

The requirements statement includes only one requirement that simply and clearly states only what shall be done, making it is easy to read and understand. To be concise, the requirements statements shall not contain any explanations, rationale, definitions, or descriptions of system use, which are used in text analysis and trade study reports, operational concept documents, user manuals, or glossaries. A link may be maintained between the requirements text and the supporting analyses and trade studies in a requirements database so that the rationale and explanations may be referenced.

Determining what constitutes one requirement is a constant struggle in developing requirements and often requires engineering judgment. An example is the requirement in FAA automation systems for a Minimum Safe Altitude Warning/Conflict Alert alarm. This alarm requires an aural alarm and a visual alarm to warn the controller about potential unsafe conditions. Therefore, the question is: Is this one requirement, or does a requirement need to be written for each condition? Multiple requirements in one paragraph are undesirable, as is the proliferation of the number of requirements without reason. Each requirement needs to be managed and verified, and as such, has an associated cost.

One decision-making approach to the question is to determine how the requirement is to be verified. In the alarm example, it is recommended to verify that the alarms work together;

therefore, any test to verify the alarms shall include both the aural and visual alarms, thus combining the aural and visual alarms into one requirement.

4.3.3.2.1.4.3 Implementation-Free

The requirement states what is required, not how the requirement needs to be met. It is recommended that the requirement state the desired result in functional and performance terms, not in terms of a solution set. It is also recommended that a requirements statement not reflect a design or implementation nor describe an operation. However, the treatment of interface requirements is generally an exception.

This characteristic of a requirement is perhaps the hardest to judge and implement. At the system level, requirements may be truly abstract or implementation-free. The system requirements have to be synthesized by a system design solution. After a trade study has been conducted between alternatives and a candidate solution has been selected, the system requirements have to be allocated to the elements defined by the system design. This incremental procedure of allocating requirements to the next lower-level elements, which is dependent on system design, leads to the observation that one level of design is the requirement at the next lower level. The conclusion is that a requirement is implementation-free at the level that it is being specified, but is a result of the design activity at the level above it.

Interface requirements are usually an exception to the implementation-free rule. Interface requirements are specified in IRDs that describe a specific design or an interface or mating part. The interface requirement shall provide complete information so that the two sides of the interface may be designed to work as specified when connected to each other.

4.3.3.2.1.4.4 Attainable (Achievable or Feasible)

The stated requirement may be achieved by one or more developed system concepts at a definable cost. This implies that at least a high-level conceptual design has been completed and cost tradeoff studies have been conducted.

This characteristic is a test of practicality of the numerical value or values set forth in a requirement. It signifies that adequate analyses, studies, and trades have been performed to show that the requirement may be satisfied by one or more concepts and that the technology cost associated with the concept(s) are reasonable within program cost constraints.

4.3.3.2.1.4.5 Complete (Standalone)

The stated requirement is complete and does not need further amplification and provides sufficient capability.

This characteristic specifies that each requirement be stated simply using complete sentences. It is recommended that each paragraph state everything required on the topic and that the requirement be capable of standing alone when separated from other requirements.

4.3.3.2.1.4.6 Consistent

The stated requirement does not contradict other requirements and is not a duplicate of another requirement. The same term is used for the same item in all requirements.

This characteristic of well-defined requirements is usually well understood and does not cause much discussion. However, in a large set of requirements that are not well organized by some clearly defined categories, it may be hard to spot duplications and inconsistencies. Therefore, organizing requirements in accordance with a standard or template is important so that inconsistencies may be identified. It is also important to maintain a glossary of program terms because the meaning of some words is domain-dependent.

4.3.3.2.1.4.7 Traceable

It is recommended that each stated requirement be developed in a way that allows it to be traced back to its source. A requirement also needs to identify related requirements (i.e., parents, children, peers) and requirements that might be impacted by changes to it.

This characteristic contributes to completeness by verifying that all requirements have a source or are allocated. It also helps to eliminate unnecessary or missing requirements.

4.3.3.2.1.4.8 Unambiguous

Each requirement shall have one, and only one, interpretation. Language used in the statement shall leave no doubt as to the intended descriptive or numeric value.

This characteristic is difficult to achieve because the English language may be unstructured and, in some cases, the same sentence may mean different things to different people. It is helpful to use standard specification language constructs and commonly used words and phases and to avoid using the commonly used words and phrases cited in Paragraph 4.3.3.2.1.1.5.3.

4.3.3.2.1.4.9 Verifiable

Each requirement shall have an identified means by which to verify that it meets the characteristics established above.

The stated requirement is not vague or general but is quantified in a manner that may be verified by one of the verification methods described in Validation and Verification (Section 4.12).

The characteristic of verifiability needs to be considered at the same time that a requirement is being defined. A requirement that is not verifiable is a problem because it involves the acceptability of the system. To be verifiable, a requirement shall be stated in measurable terms.

4.3.3.2.1.4.10 Allocatable

It is recommended that the stated requirement be allocated to component(s) within the requirements hierarchy or assigned to an organization.

This characteristic is important because it helps to eliminate requirements that are not complete, concise, and clear. If a requirement is not allocatable to the Physical Architecture, it is probably not necessary.

4.3.3.2.1.5 Characteristics of Aggregate Requirements

Aggregate requirements are a set of requirements for a system or element that specifies its characteristics in totality. Usually, these aggregates are found in specifications or Statements of Work (SOW). Characteristics of individual requirements also are applicable to aggregates.

4.3.3.2.1.5.1 Complete

The set of requirements is complete and does not need further amplification. The set of requirements has addressed all categories (Paragraph 4.3.3.2.1.4.3) of requirements and covers all allocations from higher levels.

This characteristic addresses the difficult problem of identifying requirements that are necessary but are missing from the requirements set. One approach to identify missing requirements is to walk through the Operational Concept and its associated scenarios from start to finish, then walk through the same set of scenarios and ask "what if" questions. This approach usually uncovers a new set of requirements. A second approach is to develop a checklist of topics or areas, such as a specification outline, and verify that requirements exist in each topic area or, if they do not exist, that there is a good reason for it. A third approach is to check the aggregate requirements set against a higher-level document (if one exists) to verify that all allocated requirements have been included in the set.

4.3.3.2.1.5.2 Consistent

The set of requirements has no individual requirements that are contradictory. Requirements are not duplicated, and the same term is used for the same item in all requirements.

This characteristic addresses the problem of identifying unnecessary or conflicting requirements that are inadvertently included in the set. Assigning program-unique identification to each requirement and conducting thorough reviews are ways to eliminate these requirements.

4.3.3.2.1.6 Attributes of Requirements

This section describes secondary properties or attributes of individual requirements that provide supplementary information about the requirement and its relationship to other requirements and source documents. The properties or attributes also assist in requirements management. However, these attributes are not essential in all cases.

4.3.3.2.1.6.1 Requirement Identification

Each requirement is assigned a program-unique identifier (PUI) for identification and tracking purposes. The PUI may be either numeric or alphanumeric and assigned automatically if a requirements management tool is used. The requirement identifier assists in identifying the requirement, maintaining change history, and providing traceability.

4.3.3.2.1.6.2 Level

This attribute indicates the level at which the specific requirement is applicable in the system hierarchy or Work Breakdown Structure (WBS). A level I requirement may indicate a top- or system-level requirement; a level II requirement may be a segment- or component-level requirement.

4.3.3.2.1.6.3 Requirements Category

Two categories are used to classify requirements: program and technical.

4.3.3.2.1.6.3.1 Program Requirements

Program requirements are stakeholder or user requirements imposed on vendors through contractual vehicles other than specifications, including the contract or contract SOW. Program requirements include:

- Compliance with Federal, State, or local laws, including environmental laws
- Administrative requirements (e.g., security); stakeholder/vendor relationship requirements (e.g., directives to use government facilities for specific types of work such as test); and specific work directives (e.g., directives included in the SOW and Contract Data Requirements List (CDRL))

Program requirements may also be imposed on a program by agency policy, directives, or practice.

Program requirements are different from technical requirements: They are not imposed on the system or product to be delivered but on the process to be followed by the program. Program requirements, which are managed similarly to technical requirements, need to be necessary, concise, attainable, complete, consistent, and unambiguous in the same manner as technical requirements.

4.3.3.2.1.6.3.2 Technical Requirements

Technical requirements are applicable to the system or service to be procured. Technical requirements are described in requirement documents, system specifications, and interface documentation. Types of technical requirements are described in the following paragraphs.

4.3.3.2.1.6.3.2.1 Stakeholder Requirements

Stakeholder requirements are associated with the stakeholder's intended operating practices, maintenance concepts, and desired features.

4.3.3.2.1.6.3.2.2 Operational Requirements

Operational requirements define the interfaces between the end-user and each functional system, maintenance concept and each system, and various other support and related functions or equipment.

4.3.3.2.1.6.3.2.3 Performance Requirement

Performance requirements define how well the product performs its intended function (e.g., accuracy, fidelity, range, resolution, and response times).

4.3.3.2.1.6.3.2.4 Functional Requirements

Functional requirements identify what the system may do, not how the system accomplishes it. They are based on Functional Analysis (Section 4.4).

4.3.3.2.1.6.3.2.5 Interface Requirements

Interface requirements are the physical and functional requirements associated with the product interfaces (boundary conditions). Interface development is described in Interface Management (Section 4.7).

4.3.3.2.1.6.3.2.6 Constraint Requirements

Constraint requirements are limitations or restrictions that bound the solution set.

4.3.3.2.1.6.3.2.7 Regulatory Requirements

Regulatory requirements are imposed by statutes or regulations (e.g., FARs, Occupational Safety and Health Administration (OSHA) regulations, and Environmental Protection Agency (EPA) directives).

4.3.3.2.1.6.3.2.8 Reliability, Maintainability, and Availability/Supportability

Reliability, maintainability, and availability/supportability requirements are based on the user's system readiness and mission performance requirements, physical environments, and resources (e.g., personnel, training, and facilities) available to support the mission.

4.3.3.2.1.6.3.2.9 Safety Requirements

These requirements are defined to control the effects of failure conditions, hazards, and/or safety-related functions.

4.3.3.2.1.6.3.2.10 Health Hazard Requirements

These requirements are defined to control the effects of failure conditions, hazards and health related functions.

4.3.3.2.1.6.3.2.11 Human Engineering Requirements

Human requirements define the human system interface(s).

4.3.3.2.1.6.3.2.12 Producibility Requirements

Producibility requirements define the producibility of a product that involve identifying materials, special tools, test equipment, facilities, personnel, and procedures. They identify the manufacturing technology needs, availability of critical materials, long-lead procurement requirements, and manufacturing test requirements, among other aspects.

4.3.3.2.1.6.3.2.13 Cost Requirements

Cost requirements define product budget constraints.

4.3.3.2.2 Decompose Requirements

The requirements may be decomposed to the lowest level and partitioned in such a way that integrating the partitioned requirements shall satisfy the higher-level requirement.

4.3.3.2.3 Checklist for Writing and Evaluating Requirements

The following guidelines for writing and evaluating requirements contain representative questions; however, the list is not intended to be complete and comprehensive.

4.3.3.2.3.1 Technical Considerations

- Does the requirement state a valid need?
- Is the requirement verifiable?
- Has the verification approach been identified?
- Are the necessary interface requirements stated?
- Are appropriate data (e.g., tables, figures) included?
- Are the stated references clearly applicable to the requirement?
- Is the requirement within the span of knowledge of the requirement owner?
- Does the requirement have stated values for quantities?
- Are words that imply a design avoided?

4.3.3.2.3.2 Traceability Considerations

- Are the applicable parent, child, and peer requirements identified?
- Are the source and rationale for the existence of the requirement documented?
- Is the basis for allocation identified?

4.3.3.2.3.3 Writing Considerations

- Is the requirement stated as a requirement?
- Is the requirement stated clearly and concisely?
- Does the requirement represent only one thought?
- Is the requirement stated positively?
- Is the requirement void of ambiguous terminology?
- Is the requirement grammatically correct?
- Is the requirement punctuated correctly?
- Is excessive punctuation avoided?

4.3.3.3 Task 3: Allocate Requirements

4.3.3.3.1 Allocation

The Allocate Requirements activity allocates or assigns requirements to system, personnel, or support activity components and/or appropriate organizational entities. This process verifies that the performance and verification requirements are correct and complete at each level before further allocation and decomposition, and it verifies them regarding feasibility and top-level design concept before allocation to software. The allocated requirements consist of all requirements, including the breakdown/decomposition of physical characteristics, functions, cost, schedule, reliability/maintainability parameters, and performance parameters. Mapping of these requirements identifies the owner that has Responsibility, Authority, and Accountability (RAA) for the respective requirement.

4.3.3.3.2 Application

The Allocate Requirements activity is applied iteratively when new, changed, or derived requirements are generated. One cycle through the Allocate Requirements activity is complete when the currently identified requirements have been accurately allocated to the appropriate system, personnel, or support activity component(s). Subsequent analyses, requirement decomposition, and trade studies may produce additional requirements that define the most balanced requirements allocation for the product. When a system-level requirement is allocated to more than one configuration item, the allocation process ensures that the lower-level requirements, when taken together, satisfy the system requirements.

4.3.3.3.3 Allocation Hierarchy

Typically, the requirements are allocated to components of the system hierarchy defined in the Physical Architecture provided by the Synthesis process (Section 4.5). System requirements (including test and verification requirements) are analyzed, refined, and decomposed to ensure complete functional allocation to system, personnel, or support activity components. When a system-level requirement is allocated to more than one configuration item, a process is used to ensure that the lower-level requirements, when taken together, satisfy the system-level requirement. Early allocations only designate high-level product components, as a complete design may not have been determined. As the product design matures, the identified requirements may be allocated to lower-level components in the Physical Architecture. The requirements documents below the system level are simply documents containing the requirements that have been allocated to particular product component(s). As requirements are identified and allocated at different levels of the product hierarchy, the requirements documents may be produced and formatted to fit the need at that particular level. As the requirements and system hierarchy are iteratively defined to lower levels, each requirement ultimately shall be allocated to the lowest possible level of the system component. The results of the allocation process are documented in the Requirements Allocation Matrix (RAM) described in Paragraph 4.3.4.1.1.3.

4.3.3.3.4 Hardware/Software Allocation

The requirements allocation process allocates design requirements to hardware and software. Software, hardware, and interface specifications are analyzed and refined to ensure that all requirements allocated to software and hardware are adequately addressed and that they do not include inappropriate levels of design details. Occasionally, requirements are derived from software requirements; these requirements are documented and maintained. In addition to allocating requirements to system elements, the process allocates requirements to incremental blocks and builds. The process establishes functional, performance, and verification requirements for each incremental system or software block or build.

4.3.3.3.5 Allocation Program Responsibility

Although SE does not establish program organization, the program organization shall contain elements responsible for allocating requirements and deriving design from the system specification to the software and hardware configuration items.

4.3.3.4 Task 4: Derive Requirements

4.3.3.4.1 Identify Derived Requirements

The objective of requirements derivation is to identify and express requirements that result from considering functional analysis, higher-level requirements, constraints, or processes. This results in additional clarification or amplification of higher-level requirements. These derived requirements need to be stated in measurable parameters at increasingly lower levels within the product hierarchy. Derived requirements may result from, but are not limited to:

- Regulatory policies, program policies, agency practices, and supplier capabilities.
- Environmental and safety constraints; the process translates and traces safety-specific system requirements into the software and hardware requirements baseline. Safety program requirements are also reflected in organizational standards and procedures. The process translates and traces safety-specific requirements into the system (hardware and software) baseline. The process assesses system safety program requirement tasks for applicability and incorporation into organizational standards and procedures.
- Architecture choices for performing specific system functions.
- Design decisions.
- Hardware-software interfaces not already specified in the baseline interface documentation.
- Establishment of detailed requirement values and tolerances (i.e., minimum, maximum, goal, threshold).

Impacts of derived requirements need to be analyzed progressively in all directions (parent, child, and peer) until it is determined that no additional impact is propagated. During this process, the hardware and software architecture design is reviewed for flexibility to adapt to new system requirements.

4.3.3.4.2 Capture Derived Requirements

Derived requirements are captured and treated in a manner consistent with other requirements applicable during the development stage. This activity, like overall SE, is an iterative operation, constantly refining and identifying new requirements as the product concept develops and additional details are defined. As part of the requirements derivation process, areas of the system with volatile requirements are monitored, and requirements specifications are reviewed for ambiguities with the potential of causing software sizing and timing instability and other program impacts.

4.3.3.5 Task 5: Establish Verification Methodology

In this step, a verification approach is developed for each requirement documented in the Validation Table, and the Validation Table is transformed into a VRTM. The strategy or method used to verify each requirement is specified in a Verification Requirement, and the Verification Requirements are listed in the VRTM. The VRTM defines how each requirement is to be verified, the stage in which verification is to occur, and the applicable verification levels. The verification approaches are:

- Inspection
- Analysis
- Demonstration
- Test

These methods are discussed in Validation and Verification (Section 4.12). Figure 4.12-2 is an example of a VRTM. Specific guidelines for the VRTM are included in the Test and Evaluation section of the FAST (http://fast.faa.gov/toolsets/index.htm).

4.3.3.6 Task 6: Manage Requirements Changes

This activity manages and controls requirements throughout the product's lifecycle (before and after instituting formal configuration control) by means of a defined change process. The activity identifies and controls all issues and decisions, action items, formal and informal stakeholder/program management desires/directives, and any other real or potential changes to the requirements. The activity is invoked when a new requirement is identified or a change occurs during any other activity within the Requirements Management process. The activity is a project-wide, approved approach that documents and controls the identified requirement, its appropriate attributes, its relationship(s) to other requirements, and allocation to the product of functional and/or verification hierarchies. The activity ensures that all involved stakeholders concur with the baselined requirements and any changes. The change process controls the allocation of requirements between hardware and software. This activity shall be conducted in conjunction with the Configuration Management process (Section 4.11).

This process accounts for changes to Government-Furnished Equipment (GFE) and Contractor-Furnished Equipment (CFE) safety critical items that impact development efforts. The process also accounts for changes resulting from the Verification process (Section 4.12). That is, if a test or other form of verification determines that a change in requirements is necessary, the process ensures that the change process is initiated to accomplish that change. The steps described in the following paragraphs are performed.

4.3.3.6.1 Identification

A new requirement or a change to an existing requirement is identified. The originator documents the new requirement or change to an existing requirement by providing, at minimum, the following information to the requirements analysis team:

- Statement of the requirement.
- Justification/rationale (e.g., trade study, documentation).
- Traceability, if applicable, to the parent child and/or peer requirements(s). Two-way traceability between the software requirements and the system requirements is established and maintained.
- List of other elements (e.g., physical or functional hierarchies) impacted. For example, whenever requirements change, there is a review of and an update to the hardware and software architecture design. This process ensures that the software impact for each proposed change is addressed. Software artifacts (e.g., requirements, design, code, and documentation), for example, are revised as changes to the requirements are

incorporated. In addition, software development plans and program baselines (e.g., cost and schedule) are reviewed and modified if necessary.

- Change requests and problem reports for all configuration items or units are initiated, recorded, reviewed, approved, and tracked.

4.3.3.6.2 Control

The requirements analysis team prepares and disseminates a requirements change notification as follows:

- Assign due date
- Collect and resolve conflicting responses—if not received, assume acceptance
- Place on decision authority agenda
- Present to appropriate decision authority and record the disposition

Multiple approval levels may be established, depending on management methodology, size, or project phase. If concurrence is not reached, the requirement shall be elevated to the next higher-level review board or decision authority; that is:

- Project Configuration Control Board (CCB)—Changes that impact only the project products
- Program CCB—Changes that impact projects outside of individual projects
- NAS CCB—Changes that are NAS-wide in scope or affect NAS-level requirements or architecture

4.3.3.6.3 Status Accounting

The disposition is recorded and the decision is disseminated to the involved stakeholders. At the program and NAS level, a Configuration Control Decision shall be issued. Otherwise, the project issues new/revised requirements document(s), Specification Change Notices (SCN), requirements verification document(s), and compliance report(s), as appropriate.

4.3.4 Outputs of Requirements Management

4.3.4.1 External Outputs

4.3.4.1.1 Requirements

4.3.4.1.1.1 Requirements Documents

The term “requirements documents” refers to any media that record requirements, either in hard copy or electronic form. It is a basic rule that all requirements shall be recorded, including internally generated requirements as well as those generated external to the project. The process does not allow verbal or unwritten requirements.

4.3.4.1.1.1.1 Stakeholder Requirements Documents

Standard requirements documents from an FAA stakeholder include the MNS, the iRD, and the fRD. Other organizations use the Operational Requirements Document (ORD) to communicate requirements. Stakeholders convey requirements through memoranda and other media.

4.3.4.1.1.1.2 Specifications

Specifications are a standard form of requirements documents. The technical requirements for a system and its elements are documented through a series of specifications as described in this manual. FAA-STD-005e, “Preparation of Specifications, Standards and Handbooks,” describes the requirements for preparing FAA specifications, standards, and handbooks. MIL-STD-961 is the current standard format for FAA specifications required by FAA-STD-005e. FAA specifications were prepared in the MIL-STD-490 format until recently, and some legacy specifications remain in that format. However, MIL-STD-490 specifications may continue to be used for reference. Newly prepared specifications shall be prepared in accordance with FAA-STD-005e.

4.3.4.1.1.1.2.1 Types of Specifications

The System Specification (Type A) is the single most important engineering design document, defining the system functional baseline and including the results from the needs analysis, feasibility analysis, operational requirements and the maintenance concept, top-level functional analysis, and the critical TPMs. This top-level specification leads to one or more subordinate specifications covering applicable subsystems, configuration items, equipment, software, and other system components. Although the individual specifications for a given program may assume a different set of designations, a generic approach is used here.

4.3.4.1.1.1.2.1.1 System Specification (Type A)

Type A Specification includes the technical, performance, operational, and support characteristics for the system as an entity. It includes allocation of requirements of functional areas, and it defines the various functional-area interfaces. The information derived from the feasibility analysis, operational requirements, maintenance concept, and functional analysis is covered. The Type A specification is the FAA-E-XXXX specification described in FAA-STD-005e.

The System Specification shall provide the technical baseline for the system as an entity, shall be written in performance-related terms, and shall describe design requirements in terms of “whats,” including the functions that the system is to perform and the associated metrics.

The System Specification is the requirements document used by the FAA to procure most systems. It is placed under configuration management before the system Request for Proposal (RFP) is issued.

4.3.4.1.1.1.2.1.2 Development Specification (Type B)

Type B Specification includes the technical requirements for any item below the system level where research, design, and development are accomplished. This may cover an equipment item, assembly, computer program, facility, or critical item of support. Each specification shall include the performance, effectiveness, and support characteristics that are required in evolving design from the system level down.

The Development Specification is usually produced by a system vendor in response to the

FAA-developed System Specification. It is placed under configuration management at completion of the Preliminary Design Review (PDR).

4.3.4.1.1.1.2.1.3 Product Specification (Type C)

Type C Specification includes the technical requirements for any item below the top system level that is currently in the inventory and may be procured off the shelf. This may cover standard system components (e.g., equipment, assemblies, units, cables), a specific computer program, a spare part, or a tool. The Product Specification is usually produced by a system vendor in response to the FAA-developed System Specification or to a vendor-developed Development Specification. It is placed under configuration management at completion of the PDR.

4.3.4.1.1.1.2.1.4 Process Specification (Type D) (Rarely Used in Federal Aviation Administration Procurements)

Type D Specification includes the technical requirements that cover a service that is performed on any component of the system (e.g., machining, bending, welding, plating, heat treating, sanding, marking packing, and processing).

The Process Specification is usually produced by a system vendor in response to the FAA-developed System Specification. It is created by the vendor and is rarely used in FAA procurements.

4.3.4.1.1.1.2.1.5 Material Specification (Type E) (Rarely Used in Federal Aviation Administration Procurements)

Type E Specification includes the technical requirements that pertain to raw materials, mixtures (e.g., paints, chemical compounds), or semi-fabricated materials (e.g., electrical cable, piping) that are used in the fabrication of a product.

The Material Specification is usually produced by a system vendor in response to the FAA-developed System Specification. It is created by the vendor and is rarely used in FAA procurements.

4.3.4.1.1.2 Requirements Change Notices

An SCN is a formal document specifying that a baselined document has been changed.

4.3.4.1.1.3 Requirements Allocation Matrix

The RAM allocates requirements to components and assigns responsibilities to organizations. Normally, a requirements management tool, such as Dynamic Object-Oriented Requirement System (DOORS), is used for this purpose. A RAM contains the following data:

- Text-based requirement.
- Detailed source of the requirement (i.e., person, document and paragraph number).

- Assigned team(s).
- Traceable parent and/or child requirements. Two-way traceability between the design and the requirements is established and maintained. In addition, when software is reviewed against the design, two-way traceability between the software code and design is established and maintained. Two-way requirements traceability is maintained from system specification to hardware and software configuration item specifications.
- Date of inclusion or deletion.
- Reference WBS number.
- Requirements verification method (i.e., test, analysis, inspection, demonstration).
- Allocated cost estimate, if any.
- Any CDRL item(s) associated with the requirement.

4.3.4.1.1.4 Requirements Database

Although requirements are normally provided in the hard-copy formats described above, they are also available in the original electronic format in automated tools such as DOORS.

4.3.4.1.2 Requirements Verification Compliance Document

The RVCD is output to program and project management for program control activities.

4.3.4.1.3 Verification Requirements Traceability Matrix

The VRTM is included as a part of every requirement and specification document. It provides information on the verification and traceability from a requirement to a higher-level requirement or to its ultimate source. Validation and Verification (Section 4.12) provides more information on this topic.

4.3.4.2 Internal Outputs

Internal outputs are products that are provided to other SE processes.

4.3.4.2.1 Technical Planning

4.3.4.2.1.1 Planning Criteria

Planning criteria describing planned activities for the Requirements Management process are output to the Integrated Technical Planning process (Section 4.2).

4.3.4.2.2 Functional Analysis

4.3.4.2.2.1 Mission Need Statement

The MNS is output to Functional Analysis (Section 4.4) for use as the baseline for developing the next lower-level Functional Architecture that is then used by the Requirements Management process to develop the next lower-level requirements.

4.3.4.2.2.2 Requirements

The requirements set at any stage in the requirements development process are output to the Functional Analysis process (Section 4.4) for developing the next lower-level functional analysis.

4.3.4.2.3 Synthesis

4.3.4.2.3.1 Requirements

The requirements set below the MNS are output to the Synthesis process (Section 4.5), which allocates requirements to the Physical Architecture.

4.3.4.2.4 Trade Studies

4.3.4.2.4.1 Requirements

During the Requirements Development process, alternative solutions may be proposed that require analysis by conducting trade studies. The Requirements Management process provides output requirements for analysis to the Trades Studies process (Section 4.6).

4.3.4.2.4.2 Constraints

Constraints that are developed during the Identify and Capture Requirements task may be used in a trade study and are output to the Trade Studies process (Section 4.6) in addition to requirements.

4.3.4.2.5 Interface Management

4.3.4.2.5.1 Mission Need Statement

The MNS is provided to the Interface Management process (Section 4.7) so that functional and physical interfaces may be identified and placed under management.

4.3.4.2.5.2 Requirements

Requirements are provided to the Interface Management process (Section 4.7) at all stages of requirements development so that interfaces are identified and controlled.

4.3.4.2.6 Specialty Engineering

4.3.4.2.6.1 Requirements

To perform Specialty Engineering analyses, the system under study shall be described. Requirements are a key component of any description, and they are an output to Specialty Engineering (Section 4.8).

4.3.4.2.7 Integrity of Analysis

4.3.4.2.7.1 Tools/Analysis Requirements

Requirements for tools or analysis that are needed during the Requirements Management process are output to the Integrity of Analysis process (Section 4.9) so that Analysis Criteria may be developed.

4.3.4.2.7.2 Requirements

Requirements are output to the Integrity of Analysis process (Section 4.9).

4.3.4.2.8 Risk Management

4.3.4.2.8.1 Concerns and Issues

Concerns and Issues related to accomplishing the mission objectives and satisfying Stakeholder Needs that are discovered during the Requirements Management process are provided to the Risk Management process (Section 4.10) for review and resolution.

The cumulative status of requirements as a result of previous requirements reviews regarding coverage, balance, mutual conflicts, induced constraints, and so forth are analyzed, and Concerns and Issues are identified.

In the course of performing SE, it is possible that potential requirements management problems may surface in the form of Concerns and Issues. These Concerns and Issues may take many forms, but, for the most part, they may be potential risks to the program.

4.3.4.2.8.2 Requirements

The Requirements Management process identifies requirements to Risk Management (Section 4.10) that are to be analyzed for potential risk.

4.3.4.2.9 Configuration Management

4.3.4.2.9.1 Requirements

The Requirements Management process identifies requirements to the Configuration Management process (Section 4.11) that are to be controlled.

4.3.4.2.10 Validation

4.3.4.2.10.1 Requirements

Requirements developed through the Requirements Management process are to be submitted to the Validation process (Section 4.12) to determine if they are complete, concise, and necessary.

4.3.4.2.11 Verification

4.3.4.2.11.1 Verification Requirements Traceability Matrix

The Requirements Management process expands the Validation Table into a VRTM with assigned verification methods and submits the VRTM to the Verification process (Section 4.12).

4.3.4.2.11.2 Requirements

The Requirements Management process submits requirements to be verified to the Verification process (Section 4.12).

4.3.5 Requirements Management Process Metrics

Performance of this process is measured and recorded on a regular basis. The following metrics, at minimum, may be used to evaluate process performance:

- Number of requirements, including both stakeholder-specified and project-derived
- Number of changed requirements, including both stakeholder or project-initiated
- Technology requirements, including proven, to be defined, and unknown technology
- Unclear, undefined, or ambiguous requirements
- Cycle time from requirement change initiation to decision
- Cycle time from change decision to baseline incorporation
- Percent of validated requirements to total proposed requirements

4.3.6 Automated Tools for Requirements Management

Use of an automated requirements tool for documenting requirements and related information depends on a variety of factors (e.g., size and complexity of the program, number of requirements, budget). There are multiple automated software tools in the marketplace that adequately store and retrieve the requirements and their traceability. A program's tool shall be capable of maintaining two-way traceability, from system specifications to hardware and software configuration item specifications. It shall be capable of being integrated into an overall SE tool suite so that data are seamlessly portable between applications.

For small programs, a spreadsheet may be more than adequate to document and control the requirements set. As a program grows and becomes more complex, a tool designed for requirements management may be necessary. The primary requirements tool used by the FAA and many of the FAA's systems vendors is DOORS.

4.3.6.1 Requirements Database Accessibility

The requirements information shall be accessible by all program personnel. This may be accomplished by allowing user access to the database itself or by providing availability to the documentation out of the database. A program decision shall be made concerning the availability and changeability of the requirements data. All personnel may be trained in using the requirements management tool or database, or a select group may manipulate the database and use a distribution media (e.g., intranet Web site, paper) to disseminate the information and collect comments and changes.

4.3.6.2 Requirements Tool Characteristics

It is recommended that the database be capable of identifying (i.e., in the form of attributes and relationships) and presenting (e.g., internal queries, standard and project-unique reports) the following types of information:

- **Requirements documentation—**statements of the requirements, status, requirement type, rationale, and history (including data configuration control) regarding each requirement, and the ability to present the requirements in an appropriate user-defined format (e.g., requirement documents, specifications)
- **Traceability**—linking requirements to their parent, child, and peer requirements, resulting in user-defined requirement traceability matrices
- **Allocation—**linking requirements to the product hierarchy, resulting in user-defined requirements allocation documents
- **Verification**—linking the requirement to specific verification approach attributes, resulting in requirements verification and compliance documents
- **Traceability Impact Assessment—**ability to assess the impact of proposed changes to the requirement, product, and verification hierarchies
- **Compatibility**—ability to communicate (minimum of import and export capabilities) with other automated tools

4.3.7 References

1. Blanchard, B. "System Engineering Management." (2nd ed.). John Wiley & Sons, Inc: New York, NY.
2. Defense Systems Management College. "Systems Engineering Fundamentals." Defense Systems Management College Press: Fort Belvoir, VA, 1999.
3. Grady, Jeffrey. "System Requirements Analysis." McGraw Hill: New York, NY, 1993.
4. Hooks, Ivy. "Guide for Managing and Writing Requirements." Compliance Automation, Inc: Houston, TX.
5. "IEEE Standard for Application and Management of the Systems Engineering Process." The Institute of Electrical and Electronic Engineers, New York, NY.
6. Kar, Philip, and Bailey, Michelle. "Characteristics of Good Requirements." Compliance Automation, Inc: Houston, TX.
7. MIL-STD-961D. 22 March 1995. "Department of Defense Standard Practice for Defense Specifications."
8. Mission Needs Statement.
9. National Airspace System Architecture.
10. National Airspace System Concept of Operations. Version 2.
11. Requirements Documents.
12. Sage, A., and Rouse, W. "Handbook of Systems Engineering and Management." John Wiley & Sons, Inc: New York, NY.
13. Young, R. "Effective Requirements Practices." Addison-Wesley: New York, NY.